

REMARKS

35 U.S.C. § 103 Claim Rejections

By the Office Action dated March 21, 2007, the Examiner has rejected claims 1-27 under 35 U.S.C. § 103(a) as being unpatentable over Lipkin (U.S. 2005/0154699)

5 (hereinafter "Lipkin") in view of Challenger (U.S. Patent No. 6,026,413) (hereinafter "Challenger"). In order to form a proper obviousness rejection of a claim under 35 U.S.C. § 103(a), a collection of references together must teach or suggest each element of the claim, including the relationships between the elements. If any element is not fully taught by the combined references, the rejection cannot be sustained.

10 Evaluating Lipkin in view of Challenger in this light, it is appropriate to examine the portions of Lipkin in view of Challenger that the Examiner has pointed to as teaching the claimed elements of the rejected claims.

Claims 1-15

The Examiner asserted that, for claim 1,

15

Lipkin discloses substantially all of the elements, a client-server system capable of validating cached

eXtensible Markup Language (XML) data comprising

a data store for storing XML data;

20

a server for retrieving and updating XML data in the data store to service client requests;

a transformation engine for transforming XML data into a format suitable for a client application based on a set of transformation

rules (Paragraph [1190])

25

. (See Office Action, page 2.) The Examiner then admitted that, for claim 1,

Lipkin discloses substantially all of the elements . . . except

a cache for temporarily storing transformed XML data as data objects for later reuse;

30

a cache monitor for ensuring that cached objects are validated

when changes to XML data in the data store are detected by the server; and

an object dependency mapper for automatically and continuously determining dependencies between XML data in the data store and sets of transformation rules.

(See Office Action, page 2.) The Examiner then asserted that, for claim 1,

Challenger teaches in figures 1, 4-16, 18-21, 23-28, and 30-42,

a cache for temporarily store, and

monitor the change to data,

an object dependency mapper for automatically and

continuously determining dependencies between data in store

and set of transformation rules (Abstract; col. 4, lines 6-10;

col. 10, lines 14-24; col. 29, lines 32 +).

(See Office Action, pages 2 and 3.) The Examiner finally asserted that, for claim 1, “[i]t would have been obvious to one having ordinary skill in the art at the time the invention was made to use a cache for temporarily store and monitor for improved performance.”

(See Office Action, page 3.)

Claim 1

To the extent the Examiner's language at pages 2 and 3 of the Office Action can be understood, it appears that the Examiner has asserted the following correspondence between Lipkin and Challenger and claim 1:

<u>Claim 1</u>	<u>Lipkin</u>	<u>Challenger</u>
1. A client-server system capable of validating cached <i>eXtensible Markup Language (XML)</i> data comprising:	-	<u>Challenger</u> does not teach this claim element.

a data store for storing <i>XML</i> data;	-	<u>Challenger</u> does not teach this claim element.
a server for retrieving and updating <i>XML</i> data in the data store to service client requests;	-	<u>Challenger</u> does not teach this claim element.
a transformation engine for transforming <i>XML</i> data into a format suitable for a client application based on a set of transformation rules;	-	<u>Challenger</u> does not teach this claim element.
a cache for temporarily storing transformed <i>XML</i> data as data objects for later reuse;	<u>Lipkin</u> does not teach this claim element.	<u>Challenger</u> does not teach this claim element.
a cache monitor for ensuring that cached objects are validated when changes to <i>XML</i> data in the data store are detected by the server; and	<u>Lipkin</u> does not teach this claim element.	<u>Challenger</u> does not teach this claim element.
an object dependency mapper for automatically and continuously determining dependencies between <i>XML</i> data in the data store and sets of transformation rules.	<u>Lipkin</u> does not teach this claim element.	<u>Challenger</u> does not teach this claim element.

In reviewing the cited portions of Lipkin and Challenger, however, it becomes apparent that Lipkin and Challenger have been generalized, and, in fact, does not support the position asserted by the Examiner.

a cache for temporarily storing transformed XML data as data

5 **objects for later reuse**

In particular, Lipkin and Challenger, alone or in combination, fail to teach or suggest “a cache for temporarily storing transformed XML data as data objects for later reuse”, as required by claim 1. The Examiner admitted that Lipkin fails to teach or suggest “a cache for temporarily storing transformed XML data as data objects for later reuse”.

10 (See Office Action, page 2.) Challenger discloses (1) “Local Cache[,which] . . . is a cache (or other standard object store such as a file system) which is updated by an instance of a Trigger Monitor residing on the same physical machine as the cache itself[, and] Remote Cache[, which] . . . is a cache (or other standard object store such as a file system) which is updated by an instance of a Trigger Monitor residing on a different physical machine from
15 the cache itself.” (See Challenger, column 29, lines 52-61.) Thus, Challenger fails to teach or suggest that either the Local Cache or the Remote Cache can be used “for temporarily storing transformed XML data as data objects for later reuse”. Therefore, Challenger cannot teach or suggest “a cache for temporarily storing transformed XML data as data objects for later reuse”. Therefore, Lipkin and Challenger, alone or in combination,
20 cannot teach or suggest the claim 1 element of “a cache for temporarily storing transformed XML data as data objects for later reuse”.

a cache monitor for ensuring that cached objects are validated

when changes to XML data in the data store are detected by the server

Also, Lipkin and Challenger, alone or in combination, fail to teach or suggest “a
25 cache monitor for ensuring that cached objects are validated when changes to XML data in the data store are detected by the server”, as required by claim 1. The Examiner admitted that Lipkin fails to teach or suggest “a cache monitor for ensuring that cached objects are validated when changes to XML data in the data store are detected by the server”. (See Office Action, page 2.) Challenger discloses “a cache manager 1 (which is an example of
30 an object manager) [that] determines how changes to underlying data affect the values of objects.” (See Challenger, column 8, lines 54-56.) Thus, Challenger fails to teach or

suggest that cache manager 1 can be used “for ensuring that cached objects are validated when changes to *XML* data in the data store are detected by the server”. Therefore, Challenger cannot teach or suggest “a cache monitor for ensuring that cached objects are validated when changes to *XML* data in the data store are detected by the server”.

- 5 Therefore, Lipkin and Challenger, alone or in combination, cannot teach or suggest the claim 1 element of “a cache monitor for ensuring that cached objects are validated when changes to *XML* data in the data store are detected by the server”.

an object dependency mapper for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation rules

- 10 In addition, Lipkin and Challenger, alone or in combination, fail to teach or suggest “an object dependency mapper for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation rules”, as required by claim 1. The Examiner admitted that Lipkin fails to teach or suggest “an
- 15 object dependency mapper for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation rules”. (See Office Action, page 2.) Challenger discloses an “object manager [that] maintains the data dependence information . . . [such that] [w]hen new objects are created or the data dependencies change, the object manager is responsible for updating the appropriate information.” (See
- 20 Challenger, column 4, lines 33-37.) Thus, Challenger fails to teach or suggest that the object manager can be used “for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation rules”. Therefore, Challenger cannot teach or suggest “an object dependency mapper for automatically and
- 25 continuously determining dependencies between *XML* data in the data store and sets of transformation rules”. Therefore, Lipkin and Challenger, alone or in combination, cannot teach or suggest the claim 1 element of “an object dependency mapper for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation rules”. It is therefore clear that Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 1 and, therefore, a rejection of
- 30 claim 1 under 35 U.S.C. § 103(a) would be inappropriate.

Claim 2

Since dependent claim 2 depends on independent claim 1 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 1, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 2, and, therefore, a rejection of claim 2 under 35 U.S.C. § 103(a) is inappropriate.

5 **Claim 3**

Since dependent claim 3 depends on dependent claim 2 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 2, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 3, and, therefore, a rejection of claim 3 under 35 U.S.C. § 103(a) is inappropriate.

10 **Claim 4**

Since dependent claim 4 depends on dependent claim 3 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 3, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 4, and, therefore, a rejection of claim 4 under 35 U.S.C. § 103(a) is inappropriate.

15 **Claim 5**

Since dependent claim 5 depends on dependent claim 4 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 4, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 5, and, therefore, a rejection of claim 5 under 35 U.S.C. § 103(a) is inappropriate.

20 **Claim 6**

Since dependent claim 6 depends on dependent claim 5 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 5, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 6, and, therefore, a rejection of claim 6 under 35 U.S.C. § 103(a) is inappropriate.

25 **Claim 7**

Since dependent claim 7 depends on dependent claim 6 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 6, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 7, and, therefore, a rejection of claim 7 under 35 U.S.C. § 103(a) is inappropriate.

30 **Claims 8 and 12**

Since dependent claims 8 and 12 depend on dependent claim 7 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 7, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 8 or 12, and, therefore, a rejection of claim 8 or 12 under 35 U.S.C. § 103(a) is

5 inappropriate.

Claim 9

Since dependent claim 9 depends on dependent claim 8 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 8, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 9, and, therefore, a rejection of claim 9 under 35 U.S.C. § 103(a) is inappropriate.

10

Claims 10 and 11

Since dependent claims 10 and 11 depend on dependent claim 9 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 9, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 10 or 11, and, therefore, a rejection of claim 10 or 11 under 35 U.S.C. § 103(a) is inappropriate.

15

Claims 13, 14, and 15

Since dependent claims 13, 14, and 15 depend on dependent claim 12 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 12, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 13, 14, or 15, and, therefore, a rejection of claim 13, 14, or 15 under 35 U.S.C. § 103(a) is inappropriate.

20

Claims 16-21

The Examiner asserted that, for claim 16,

25

Lipkin discloses substantially all of the elements, a client-server system capable of validating cached

eXtensible Markup Language (XML) data comprising

a data store for storing XML data;

30

a server for retrieving and updating XML data in the data store to service client requests;

a transformation engine for transforming XML data into a format suitable for a client application based on a set of transformation rules (Paragraph [1190])

5 . (See Office Action, page 2.) The Examiner then admitted that, for claim 16,

Lipkin discloses substantially all of the elements . . . except a cache for temporarily storing transformed XML data as data objects for later reuse;

10 a cache monitor for ensuring that cached objects are validated when changes to XML data in the data store are detected by the server; and

an object dependency mapper for automatically and continuously determining dependencies between XML data in
15 the data store and sets of transformation rules.

(See Office Action, page 2.) The Examiner then asserted that, for claim 16,

20 Challenger teaches in figures 1, 4-16, 18-21, 23-28, and 30-42, a cache for temporarily store, and monitor the change to data, an object dependency mapper for automatically and continuously determining dependencies between data in store and set of transformation rules (Abstract; col. 4, lines 6-10;
25 col. 10, lines 14-24; col. 29, lines 32 +).

(See Office Action, pages 2 and 3.) The Examiner finally asserted that, for claim 16, “[i]t would have been obvious to one having ordinary skill in the art at the time the invention was made to use a cache for temporarily store and monitor for improved performance.”

30 (See Office Action, page 3.)

Claim 16

To the extent the Examiner's language at pages 2 and 3 of the Office Action can be understood, it appears that the Examiner has asserted the following correspondence between Lipkin and Challenger and claim 16:

Claim 16	<u>Lipkin</u>	<u>Challenger</u>
16. In a client-server computing system having a cache and storing <i>eXtensible Markup Language (XML)</i> data as data objects, a method for determining invalid cached objects comprising:	-	<u>Challenger</u> does not teach this claim element.
transforming <i>XML</i> data into a format suitable for a client application based on a set of transformation rules;	-	<u>Challenger</u> does not teach this claim element.
determining dependencies between cached objects and <i>XML</i> data related to the cached objects;	<u>Lipkin</u> does not teach this claim element.	<u>Challenger</u> does not teach this claim element.
monitoring updates to the related <i>XML</i> data; and	<u>Lipkin</u> does not teach this claim element.	<u>Challenger</u> does not teach this claim element.
determining the cached objects that are affected by changes to the related <i>XML</i> data based on the dependencies.	<u>Lipkin</u> does not teach this claim element.	<u>Challenger</u> does not teach this claim element.

In reviewing the cited portions of Lipkin and Challenger, however, it becomes apparent that Lipkin and Challenger have been generalized, and, in fact, does not support the position asserted by the Examiner.

determining dependencies between cached objects and XML

5 **data related to the cached objects**

In particular, Lipkin and Challenger, alone or in combination, fail to teach or suggest “determining dependencies between cached objects and XML data related to the cached objects”, as required by claim 16. The Examiner admitted that Lipkin fails to teach or suggest “an object dependency mapper for automatically and continuously determining
10 dependencies between XML data in the data store and sets of transformation rules”. (See Office Action, page 2.) Thus, the Examiner admitted that Lipkin fails to teach or suggest “determining dependencies between cached objects and XML data related to the cached objects”. Challenger discloses an “object manager [that] maintains the data dependence information . . . [such that] [w]hen new objects are created or the data dependencies
15 change, the object manager is responsible for updating the appropriate information.” (See Challenger, column 4, lines 33-37.) Thus, Challenger fails to teach or suggest that the object manager can be used for “determining dependencies between cached objects and XML data related to the cached objects”. Therefore, Challenger cannot teach or suggest “determining dependencies between cached objects and XML data related to the cached
20 objects”. Therefore, Lipkin and Challenger, alone or in combination, cannot teach or suggest the claim 16 element of “determining dependencies between cached objects and XML data related to the cached objects”.

monitoring updates to the related XML data

Also, Lipkin and Challenger, alone or in combination, fail to teach or suggest
25 “monitoring updates to the related XML data”, as required by claim 16. The Examiner admitted that Lipkin fails to teach or suggest “a cache monitor for ensuring that cached objects are validated when changes to XML data in the data store are detected by the server”. (See Office Action, page 2.) Thus, the Examiner admitted that Lipkin fails to teach or suggest “monitoring updates to the related XML data”. Challenger discloses “a
30 cache manager 1 (which is an example of an object manager) determines how changes to underlying data affect the values of objects.” (See Challenger, column 8, lines 54-56.)

Thus, Challenger fails to teach or suggest that cache manager 1 can be used for “monitoring updates to the related *XML* data”. Therefore, Challenger cannot teach or suggest “monitoring updates to the related *XML* data”. Therefore, Lipkin and Challenger, alone or in combination, cannot teach or suggest the claim 16 element of “monitoring updates to the related *XML* data”.

determining the cached objects that are affected by changes to the related *XML* data based on the dependencies

In addition, Lipkin and Challenger, alone or in combination, fail to teach or suggest “determining the cached objects that are affected by changes to the related *XML* data based on the dependencies”, as required by claim 16. The Examiner admitted that Lipkin fails to teach or suggest “an object dependency mapper for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation rules”. (See Office Action, page 2.) The Examiner also admitted that Lipkin fails to teach or suggest “a cache monitor for ensuring that cached objects are validated when changes to *XML* data in the data store are detected by the server”. (See Office Action, page 2.) Thus, the Examiner admitted that Lipkin fails to teach or suggest “determining the cached objects that are affected by changes to the related *XML* data based on the dependencies”. Challenger discloses an “object manager [that] maintains the data dependence information . . . [such that] [w]hen new objects are created or the data dependencies change, the object manager is responsible for updating the appropriate information.” (See Challenger, column 4, lines 33-37.) Challenger also discloses “a cache manager 1 (which is an example of an object manager) determines how changes to underlying data affect the values of objects.” (See Challenger, column 8, lines 54-56.) Thus, Challenger fails to teach or suggest that either the object manager or cache manager 1 can be used for “determining the cached objects that are affected by changes to the related *XML* data based on the dependencies”. Therefore, Challenger cannot teach or suggest “determining the cached objects that are affected by changes to the related *XML* data based on the dependencies”. Therefore, Lipkin and Challenger, alone or in combination, cannot teach or suggest the claim 16 element of determining the cached objects that are affected by changes to the related *XML* data based on the dependencies”. It is therefore clear that Lipkin and Challenger, alone or in combination, cannot teach or

suggest each element of claim 16 and, therefore, a rejection of claim 16 under 35 U.S.C. § 103(a) would be inappropriate.

Claims 17, 18, and 19

Since dependent claims 17, 18, and 19 depend on independent claim 16 and since
5 Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 16, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 17, 18, or 19 and therefore, a rejection of claim 17, 18, or 19 under 35 U.S.C. § 103(a) is inappropriate.

Claim 20

10 Since dependent claim 20 depends on dependent claim 19 since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 19, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 20, and therefore, a rejection of claim 20 under 35 U.S.C. § 103(a) is inappropriate.

Claim 21

15 Since dependent claim 21 depends on dependent claim 20 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 20, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 21, and therefore, a rejection of claim 21 under 35 U.S.C. § 103(a) is inappropriate.

Claims 22-27

Claim 22

20 Since claim 22, as amended, is the computer program product version of claim 16 with similar elements as claim 16, since the Examiner has asserted arguments against Claim 16 that are similar to the arguments asserted by the Examiner against claim 16, and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each
25 element of claim 16, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 22 for similar reasons that Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 16, and therefore, a rejection of claim 22 under 35 U.S.C. § 103(a) is inappropriate.

Claims 23, 24, and 25

30 Since dependent claims 23, 24, and 25 depend on independent claim 22 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of

claim 16, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 23, 24, or 25, and therefore, a rejection of claim 23, 24, or 25 under 35 U.S.C. § 103(a) is inappropriate.

Claim 26

5 Since dependent claim 26 depends on dependent claim 25 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 25, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 26, and therefore, a rejection of claim 26 under 35 U.S.C. § 103(a) is inappropriate.

Claim 27

10 Since dependent claim 27 depends on dependent claim 26 and since Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 26, Lipkin and Challenger, alone or in combination, cannot teach or suggest each element of claim 27, and therefore, a rejection of claim 27 under 35 U.S.C. § 103(a) is inappropriate.


Conclusion

15 It is therefore clear that claims 1-27 comply with the requirements of 35 U.S.C. §§ 101, 102, 103, and 112. The application is therefore in condition for allowance. Early notification to that effect is respectfully solicited.

 In the event that any issue remains unresolved, the Examiner is invited to telephone the undersigned at 408-927-3377.

20

Respectfully Submitted,



Date: May 21, 2007

Leonard T. Guzman

25

Reg. No. 46,308

IBM Almaden Research Center
650 Harry Road
C45A/J2B

30 San Jose, CA 95120

Phone Number: 408-927-3377

Facsimile Number: 408-927-3375